SHARED SECRET USAGE FOR BOOTSTRAPPING

Field of the Invention

[0001] The present invention relates to techniques in communication systems for authentication, and particularly but not exclusively to bootstrapping techniques.

Background to the Invention

[0002] So-called third generation (3G) mobile communication networks are currently being deployed. In Europe such networks conform to various standards formalised by the third generation partnership project (3GPP), which has a number of versions, generally known as releases.

[0003] In 3GPP there has been proposed (3GPP TS 33.220) an authentication infrastructure. This infrastructure may be utilised to enable application functions in the network side and on the user side to communicate in situations where they would not otherwise be able to do so. This functionality is referred to as "bootstrapping of application security", or more generally simply as "bootstrapping".

[0004] The general principles of bootstrapping are that a generic bootstrapping server function (BSF) allows a user equipment (UE) to authenticate therewith, and agree on session keys. Such authentication is preferably based on authentication and key agreement (AKA). By running AKA, the mobile terminal and the network mutually authenticate each other and agree on keys, specifically a confidentiality key (CK) and an integrity key (IK). After this authentication, the UE and an operator-controlled network application function (NAF), which may also be referred to as a service provider, may run some application specific protocol where the authentication of messages is based on the session keys agreed between the UE and the BSF.

[0005] The bootstrapping function is not intended to be dependent upon any particular network application function. The server implementing the bootstrapping function must be trusted by the home operator to handle authentication vectors.

Network application functions in the operators home network are currently proposed to be supported, but this does not preclude the possibility of support of network application functions in a visited network, or even in a third network.

[0006] In the proposals for implementation of bootstrapping techniques, it is proposed that the UE sends a service request to a NAF. The NAF must then communicate with the BSF in order to retrieve the key(s) required for authentication with the UE. The need for the NAF to communicate with the BSF delays the reaction time for a NAF access, such that a user may experience a delay.

[0007] It is an aim of the invention to provide an improved technique, which addresses the above-stated problem.

Summary of the Invention

[0008] According to the present invention there is provided a method of providing authentication information to a network application function comprising; receiving a request from a user equipment to provide authentication information to at least one network application function, and providing said authentication information to at least one network application function.

[0009] The request may be received from a user equipment to enable access to the network application function by the user equipment.

[0010] The request may include at least one network application function identity. The request may include an identifier of at least one type of network application function.

[0011] The authentication information may be bootstrapping information. The request may be received by a bootstrapping function. The method may further include obtaining said bootstrapping information.

[0012] The step of obtaining the bootstrapping information may comprise establishing an authentication procedure between the user equipment and the boot strapping function.

[0013] The authentication procedure between the user equipment and the boot strapping function may establish a shared key.

[0014] The bootstrapping information provided to the at least one network application function may be based on the shared key.

[0015] The bootstrapping information provided to the at least one network application function may include a transaction identifier. The bootstrapping information provided to the at least one network application function may include subscriber profile information.

[0016] The shared key bootstrapping information that may be provided to the network application function may be network application function specific. The step of obtaining bootstrapping information may include retrieving information from a home subscriber server.

[0017] In a further aspect there may be provided a method of providing bootstrapping information to a network application function entity to enable access by a user equipment, comprising; generating a request from a user equipment to provide bootstrapping information to at least one network application function entity; receiving said request at a bootstrapping functional entity; determining said bootstrapping information at said functional entity; and transmitting said determined bootstrapping information to the at least one network application function entity.

[0018] Said bootstrapping information may be determined in dependence on a transaction identifier included in said request.

[0019] The request may include at least one network application function identity. The request may include at least one type of network application function.

[0020] Said bootstrapping information may be determined by performing a bootstrapping operation.

[0021] In an aspect the invention provides an authentication entity including: receiving means for receiving a request from a user equipment to provide authentication information to at least one network application function entity; and transmitting means for transmitting said authentication information to the at least one network application function entity.

[0022] The authentication entity may further include determining means for determining at least one network application function entity identity from said request.

[0023] The authentication information may be bootstrapping information, the authentication entity comprising a bootstrapping functional entity. The authentication entity may further include determining means for determining said bootstrapping information.

[0024] The determining means for determining the bootstrapping information may comprise authentication means for establishing an authentication procedure between the user equipment and the boot strapping function.

[0025] The authentication means may establish a shared key. The bootstrapping information provided to the at least one network application function entity may be based on the shared key. The bootstrapping information provided to the at least one network application function entity may include a transaction identifier. The bootstrapping information provided to the at least one network application function entity may include subscriber profile information.

[0026] In an aspect the invention provides a user equipment adapted to transmit a request to an authentication entity to provide authentication information to at least one network application function entity. The request may be transmitted to a bootstrapping functional entity. The request may include one of a transaction identifier or a network application function identity.

[0027] In an aspect the invention provides a network application function entity adapted to receive unsolicited authentication information from an authentication function entity. Said authentication information may include bootstrapping information. Said authentication information may be associated with a user equipment. The authentication information may be received from an authentication function, the authentication information being established between said authentication function and the user equipment. The receipt of the authentication information from an authentication function entity may be responsive to a request by a user equipment to the authentication function entity.

[0028] In a further aspect, the invention provides a communication system including at least one user equipment and at least one network application functional entity, the system further including a bootstrapping functional entity, wherein the user equipment includes means to transmit a request to push authentication information to at least one network application function, the bootstrapping functional entity includes: receiving means for receiving the request from the user equipment, and transmitting means for transmitting said authentication information to the at least one network application function entity, and the at least one network application function includes means adapted to receive unsolicited bootstrapping information from the bootstrapping functional entity.

Brief Description of the Figures

[0029] The invention is described with regard to particular exemplary embodiments by way of reference to the accompanying drawings, in which:

[0030] Fig. 1 illustrates an exemplary architecture for implementing embodiments of the invention;

[0031] Fig. 2 illustrates the procedure followed in an exemplary embodiment of the invention; and

[0032] Fig. 3 illustrates the signalling exchange in an exemplary embodiment of the invention.

Description of Preferred Embodiment

[0033] The invention is described herein by way of reference to an exemplary embodiment. In particular, the invention is described in the context of the implementation of bootstrapping techniques in a 3GPP system architecture. The invention, however, is not limited to specifics of the embodiment described herein. For example, the invention may be applied in 3GPP2 system architectures.

[0034] Referring to Fig. 1, there is illustrated an exemplary network architecture overview for describing the bootstrapping process in accordance with the invention. The network architecture includes a user equipment (UE) 100, at least one network application function (NAF) 102, a bootstrapping server function (BSF) 104, and a home subscriber system (HSS) 106. The BSF 104 and HSS 106 form part of a home mobile network operator (MNO) 108. The UE 100 connects into the MNO 108 in accordance with well-known mobile communication techniques, which are outside the scope of the present invention. The NAF 102 may be provided in a further separate network.

[0035] The NAF 102 is hosted in a network element, preferably under the control of the MNO 108, and the BSF is also preferably hosted in a network element under the control of the MNO 108. Thus, for practical purposes, each of the NAF 102 and the BSF 104 may be considered to be a network element.

[0036] As illustrated in Fig. 1, the UE 100 communicates with the NAF 102 on a Ua interface 110. The UE 100 communicates with the BSF 104 on a Ub interface 112. The NAF 102 communicates with the BSF 104 on a Zn interface 114. The BSF 104 communicates with the HSS 106 on a Zh interface 116.

[0037] The principle of bootstrapping is that the user equipment and the bootstrapping function mutually authenticate each other, preferably using the AKA protocol, and agree on session keys that are afterwards applied between the user equipment and an operator-controlled network application function. The key material is generated specifically for each network application function independently. After the bootstrapping operation has been completed, the user equipment and the operator-controlled network application function may run some specific protocol where the authentication of messages will be based on those keys generated during the mutual authentication between the user equipment and the bootstrapping server function. The keys are thus used for authentication and integrity protection, and preferably also for confidentiality. The network application function is then able to acquire the shared key material established between the user equipment and the bootstrapping server function.

[0038] The communication interface 112 supports the bootstrapping authentication and key agreement protocol, to provide the mutual authentication and key agreement between the UE 100 and the BSF 104. This protocol is preferably based on the 3GPP AKA protocol.

[0039] The interface 116 allows the BSF 104 to fetch any required authentication information and subscriber profile information from the HSS 106. The interface 110 supports the application protocol which is secured using the session keys agreed between the UE 100 and the BSF 104, based on the protocol supported by the interface 112. The interface 114 is used by the NAF 102 to fetch the key material agreed in the protocol supported on the interface 112 from the BSF 104. The interface 114 may also be used to fetch subscriber profile information from the BSF 104.

[0040] With further reference to the flow diagram of Fig. 2 and the signalling diagram of Fig. 3, a preferred embodiment for the transfer of bootstrapping information in accordance with the invention is described.

[0041] The invention allows for the UE 100 to trigger the BSF 104 to "push" authentication information toward one or more NAFs 102. Thus, in accordance with the invention, the BSF provides authentication information to one or more NAFs without a request being made from the NAF to the BSF. The invention does not propose any modification to existing authentication protocols/techniques, or new authentication protocols/techniques. The invention may be applied in conjunction with any existing or proposed authentication protocols/techniques.

[0042] Referring to Fig. 2, in a step 200 the UE 100 prepares to trigger a push operation from the BSF 104 to the NAF 102. It should be noted that in practice the UE 100 may trigger the push operation to a plurality of NAFs.

[0043] In an embodiment, the UE 100 prepares a list of network application function identities (NAF_ID). This is represented by step 202 in Fig. 2. The network application function identities are known by the BSF 104, such that the BSF can identify the NAFs to which bootstrapping information is to be pushed once the bootstrapping information is established.

[0044] The UE 100 may also, in embodiments, trigger the BSF 104 to push bootstrapping information towards one or more NAFs 102 without the BSF performing a full bootstrapping operation. In such an embodiment, a transaction identifier (TID) from a previous bootstrapping procedure is utilised, as represented by step 204 in Fig. 2. The transaction identifier identifies one bootstrapping operation, i.e. an earlier bootstrapping operation. The use of the transaction identifier requires one or more network application function identities to be present in the request sent by the UE. The transaction identifier identifies

a previous transaction to the BSF, and the BSF can access bootstrapping information obtained and used for that previous transaction.

[0045] As represented by step 206, the bootstrap operation is begun by the UE 100 transmitting a bootstrap request toward the BSF 104. The bootstrap request signal is represented by signal 302 in Fig. 3. The content of the request message 302 is dependent upon whether the optional steps 202 and 204 are implemented. As a minimum, the request message 302 must include an IMPI (IP multimedia private identity). This uniquely identifies the UE 100.

[0046] Where optional step 202 is implemented, the request may contain one or more network application function identities, as denoted by NAF_ID*, the asterisk denoting that zero or more NAF_IDs are present.

[0047] Where optional step 204 is implemented, the request may include a transaction identifier, as denoted by TID? The designation "?" denotes that the transaction identifier TID is optional. If the request includes a transaction identifier, then it must include at least one network application function identity NAF_ID to which bootstrapping information is to be pushed. If the request does not include a transaction identifier (which inherently identifies at least one network application function identity), it may include at least one network application function identity from step 202.

[0048] Where there is no transaction identifier and no NAF_IDs included in the request, then the request from the UE does not result in the pushing of bootstrapping information to a NAF. Instead, a bootstrapping operation is simply carried out.

[0049] As illustrated by step 208 in Fig. 2, on receipt of the request message the BSF 104 determines whether the message contains a transaction identifier. If the message does contain a transaction identifier, then in a step 210 the BSF 104 determines whether such transaction identifier is valid. For example, the BSF 104 may determine an old transaction identifier to be invalid, as too much

time has elapsed since that transaction. Other checks on the validity of the transaction identifier may be performed. If the transaction identifier is determined to be valid, then the bootstrapping information used in that previous transaction, stored by the BSF 104, may be re-used. After step 210 the bootstrapping information is then ready to be pushed to the NAFs identified by the NAF_IDs in the transaction identifier, as discussed further hereinbelow, and the process moves on to step 216.

[0050] If in step 208 it is determined that there is no transaction identifier present, or if in step 210 it is determined that the transaction identifier is not valid, then the process moves to step 214, and a bootstrap operation is performed. On successful completion of the bootstrap operation, in a step 215 the bootstrap information is established.

[0051] Thereafter, in step 212 of the described embodiment, the BSF 104 determines whether the request message 302 includes any network application function identities. If any such identities are present, then the NAF-IDs are provided in step 216 to push the bootstrapping information to such network application functions. NAF identities may be present if they are specifically included in the request in step 202. Alternatively the NAF identities may be present as a result of a transaction identifier included in step 204. Thus even if the transaction identifier is determined not to be valid, the NAF identifiers associated therewith may still be used. As further described herein below, the NAF identifiers may be provided by further alternative techniques.

[0052] If step 212 determines that there are no NAF_IDs present, then the process simply moves to step 217 and the UE is notified that no push operation has been performed.

[0053] In step 214, the BSF 104 performs a conventional bootstrap operation. This conventional bootstrap operation is illustrated in Fig. 3 by messages 304,

306, 308, 310. The completion of the bootstrapping operation is represented by the establishment of the bootstrapping information in step 215.

[0054] For bootstrapping, the BSF 104 transmits a MAR (IMPI) message 304 to the HSS 106. The HSS 106 returns a MAA (AV+, profile) message 306. The signal exchange 304 and 306 allows the BSF 104 to retrieve the user profile for the UE 100, and an authentication vector for the UE 100. The authentication vector comprises a plurality of elements, as is known in the art, and is represented by the notation "AV+". "AV+" denotes one or more attribute values (AVs). The authentication vector includes RAND, AUTN, XRES, CK key and IK key. The signal exchange 304 and 306 may not be needed if the BSF already has the authentication vectors for the UE.

[0055] The BSF 104 then returns an " 401 unauthorised (RAND, AUTN delivered)" message 308 to the UE 100 i.e. the BSF 104 forwards the RAND and AUTN to the UE. The XRES, CK and IK are not forwarded. This message demands (or challenges) the UE 100 to authenticate itself.

[0056] Responsive to the message 308, the UE 100 calculates the message authentication code (MAC) so as to verify the challenge from the authenticated network. The UE calculates CK, IK, and RES. As a result, the session keys CK and IK are present in both the UE 100 and the BSF 104. The UE 100 then returns a "GET (RES used)" message 310, being a further request message. The further request message, 310, includes the Digest AKA RES as the response to the BSF challenge. If the RES contained in this message, as calculated by the UE 100, matches the XRES in the AV provided by the HSS to the BSF, then the UE is authenticated.

[0057] The key material Ks is generated, separately, in both the UE 100 and the BSF 104, by concatenating CK and IK. The key material Ka is then used to secure the Ua interface 110.

[0058] Referring to Fig. 2, in a step 216 the bootstrapping information determined is then provided to one or more NAFs 102 in a push operation. It should be noted that the NAFs may be identified in a number of ways. The NAFs may be identified by a valid TID, following steps 208 and 210, and without the need for any bootstrapping operation. The NAFs may be identified by an invalid TID, following step 210, but with the need for a bootstrapping operation. The NAFs may be identified by explicit NAF-IDs in step 212.

[0059] Thus any NAFs identified by the transaction identifier following step 210 have the bootstrapping information pushed thereto. Similarly any NAFs having their identities included in the request message, following step 212, have the bootstrapping information returned from the bootstrap step 214 pushed thereto.

[0060] In the push operation, the BSF 104 provides to the identified NAF(s) 102 the key material for the UE, which the UE has requested to be pushed to the NAF(s). The NAF(s) then derive the keys required to protect the protocol used over the Ua interface 110 in the same way as the UE 100 did.

[0061] IN general, the invention is not limited to any particular technique for the provision of the NAF identifiers. IN step 212, any NAF identifiers present are determined. Whilst the NAF identifiers may be provided in the request 302, for example, they may also be provided in the bootstrapping phase, for example at the end of the bootstrapping phase in the HTTP Digest AKA message 310.

[0062] The transmission of the bootstrapping information from the BSF 104 to one or more NAFs 102 is identified in Fig. 3 by message 312. The bootstrapping information preferably comprises a transaction identifier, a NAF specific shared secret, and an optional subscriber profile information ("prof_naf"). The NAF specific shared secret, denoted Ks_naf, is the authentication key established between the UE 100 and the BSF 104, and modified for specific use for communications between the UE 100 and the

specific NAF. Ks_naf is derived from Ks by using a parameter. Ks is the master key, and Ks_naf is a NAF specific key. The bootstrapping information transmitted to each NAF is thus unique to that NAF, in accordance with the specific shared secret Ks_naf for that NAF. As mentioned hereinabove, the nature of the information provided to any NAF as a result of the push operation in accordance with this invention is not modified.

[0063] The messages 312 and 314 are only exchanged if specific NAF_IDs are included in the original request message from the UE 100. This is described further below. If a transaction identifier is present in the request from the UE, then there must be one or more NAF identifiers present. If such NAF identifiers are present, then messages 312 and 314 are exchanged. This is represented by the flow of steps 208 to 210 to 216 in Fig. 2.

[0064] If the transaction identifier is not valid, for example because it is too old, then bootstrapping is performed in messages 304 to 310. This is represented in Fig. 2 by message flow 208 to 210 to 214. This also requires the NAF_ID associated with the TID to be provided to step 216 for the push operation. It should be noted that if no transaction identifier is present in step 208, then the method reverts to step 214. If there are no transaction identifiers present in step 208, then bootstrapping is performed in steps 214 and 215, and the NAF_IDs for pushing – if any - supplied in step 216.

[0065] Thus a push operation only occurs where at least one NAF_ID is present. In the absence of a NAF_ID, a bootstrapping operation is performed in step 214, but no push occurs, since no NAF is identified for a push to be made to in step 212 .

[0066] After transmission of the bootstrapping information to the one or more NAFs 102 as illustrated by message 312 and step 216, the NAFs 102 transmit an acknowledgement message back to the BSF 104. This acknowledges, for a

given NAF, that the bootstrapping information was received and has been stored. This is represented by step 218 in Fig. 2.

[0067] In a step 220, the UE 100 is notified of the NAFs to which the bootstrapping information has been pushed. As shown in Fig. 3, a HTTP response "200 OK" message 316 is sent to the UE 100.

[0068] It is envisaged that in one embodiment in the event that one or more NAFs do not respond with an acknowledgement, then a list of those NAFs which did positively respond with an acknowledgement is returned to the UE 100. The message 316 preferably includes a transaction identifier.

[0069] Alternatively, the response to the UE may contain no indication as to whether the push operation was successful or not.

[0070] If a full bootstrapping procedure was carried out, the response may include the new transaction identifier.

[0071] Thereafter, as represented by step 222 in Fig. 2, the user equipment may contact any NAF 102, and the NAF 102, having bootstrapping information pushed thereto, does not need to access such information from the BSF 104. As such the response time by the NAF 102 responsive to an access from the UE 100 is shortened. The UE and the NAF share the keys required to protect the Ua interface, and hence there is no need for the NAF to retrieve any key(s) from the BSF. If the NAF is a NAF to which keys have not been pushed, then a conventional operation to perform bootstrapping may be performed.

[0072] In the preferred embodiment, the bootstrapping information is pushed to network application functions which are specifically identified. That is, the bootstrapping information is pushed to those network application functions having their network application function identities provided to the BSF 104. The network application function identity, NAF_ID, is preferably in a format which is easily discovered or known by the UE 100, so that the UE 100 can

include such identities in a request to push information. It is also important that the network application function identities uniquely identify an NAF. For example, the FQDN (Fully Qualified Domain Name) uniquely identifies an NAF and may be easily discovered or known by the UE.

[0073] It is envisaged in an alternative embodiment that rather than providing the BSF 104 with specific NAF identities, the UE 100 may provide the BSF 104 with an identity of NAF types. The bootstrapping information may then be pushed to all NAFs of that type by the BSF 104, responsive to an appropriate request from the UE 100. A single identifier, identifying a NAF type, may thus result in a push to a plurality of NAFs.

[0074] In the foregoing description with relation to Fig. 2, it is shown that the BSF performs determination of a transaction identifier in step 208, and determination of NAF_IDs in step 212, and bootstrapping in step 214. Where a transaction identity is provided, there may be no requirement to perform a bootstrap operation in step 214, since the bootstrap information associated with the transaction identifier received is suitable to be used. Thus, in an embodiment, only steps 208 and 210 may be performed if a valid transaction identifier is thereby identified. If no transaction identifier is identified, then the bootstrap operation is preferably performed in step 214 in order to obtain the necessary bootstrap information.

[0075] Thus, as described, the invention provides a technique by which the user equipment requests the bootstrap function to push bootstrap information to selected network application functions. The bootstrap information is then available in those network application functions when the user equipment makes a direct access to any such network application function. As such, the network application functions do not have to dynamically access the bootstrap function in order to retrieve the bootstrap information responsive to an access from the user equipment.

[0076] The unsolicited push of bootstrapping information to selected NAFs simplifies the procedures NAF needs to do during shared key usage over the Ua interface. For example, it simplifies shared key TLS(Transport Layer Security) usage, since when the UE establishes the connection using shared key TLS between it and a NAF, the NAF would already have the related session ID and master key in its TLS cache. This would remove the need for an "active" session cache functionality.

[0077] Although the invention has been described herein by way of reference to a particular exemplary embodiment implemented in a 3GPP architecture suitable for implementation of the AKA protocol, the invention is not limited in its applicability to such an environment. More generally, the invention may be utilised in any network arrangement where access to a function is dependent upon that function retrieving authentication information.

[0078] It should be noted that the term 'network application function' has no special meaning. It is used to refer to a function or entity which provides or supports an application to which a user equipment may require access.

[0079] Various adaptations and modifications to the invention as described herein will be apparent to one skilled in the art, the scope of the invention being defined by the appended claims.